University of Colombo, Sri Lanka

University of Colombo School of Computing

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2024 — 3^{rd} Year Examination — Semester 6

IT6505 (R) — Middleware Architecture (Repeat Paper)

Structured Question Paper

(2 Hours)

To be completed by the candidate

Index Number | | | | | | | |

Important Instructions

- The duration of the paper is **2 hours**.
- The medium of instructions and questions is English. Students should answer in the medium of English language only.
- This paper has **4 questions** on **14 pages**. Answer **all** questions.
- All questions carry **equal** marks.
- Write your answers **only on the space provided** on this question paper.
- Do not tear off any part of this question paper. Under no circumstances may this paper (or any part of this paper), used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper. If a page or part of a page is not printed, please inform the supervisor/invigilator immediately.
- Any electronic device capable of storing and retrieving text, including electronic dictionaries, smartwatches, and mobile phones, is not allowed.
- Calculators are **not allowed**.
-

To be completed by the examiners

1	
2	
3	
4	
Total	

Index No …….

1)

(a) List three (03) benefits of using a layered network model in a typical protocol stack.

(6 marks)

ANSWER IN THIS BOX

(b) Explain the difference between **fault-tolerance** and **fault-resilience** in distributed computing.

(4 marks)

ANSWER IN THIS BOX

(c) In which ways can a distributed system be considered less advantageous than a centralised system in relation to complexity in design and maintenance?

(5 marks)

ANSWER IN THIS BOX

(d) List and briefly explain three types of system failure models.

(6 marks)

ANSWER IN THIS BOX

(e) List four (04) characteristic features of a distributed system.

(4 marks)

ANSWER IN THIS BOX

2)

(a) Define the term *middleware* and explain its purpose.

(5 marks)

ANSWER IN THIS BOX

(b) Can middleware be used in non-distributed systems? Explain.

(4 marks)

ANSWER IN THIS BOX

(c) Distinguish between an *operating system* and *middleware* in three (03) aspects.

(4 marks)

ANSWER IN THIS BOX

(d) Explain the use and operation of Object-Oriented Middleware.

(5 marks)

ANSWER IN THIS BOX

(e) Explain the functionality of the code below.

(**7 marks**)

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
import java.util.HashMap;
import java.util.Map;

public class StudentImpl extends UnicastRemoteObject implements Student {
    private Map<Integer, String> studentData;

    protected StudentImpl() throws RemoteException {
        super();
        studentData = new HashMap<>();
        studentData.put(1, "John Doe, Age: 21, Course: Computer Science");
        studentData.put(2, "Jane Smith, Age: 22, Course: Mathematics");
    }

    @Override
    public String getStudentDetails(int id) throws RemoteException {
        return studentData.getOrDefault(id, "Student not found.");
    }

    @Override
```

```
    public String addStudent(int id, String details) throws RemoteException {
        if (studentData.containsKey(id)) {
            return "Student ID already exists!";
        }
        studentData.put(id, details);
        return "Student added successfully.";
    }
}
```

ANSWER IN THIS BOX

3.

(a) What are the eight (08) elements of middleware?

(08 marks)

ANSWER IN THIS BOX

(b) What is meant by marshalling in Remote procedure call (RPC)?

(4 marks)

ANSWER IN THIS BOX

(c) What is serialization and how is it related to marshalling?

(4 marks)

ANSWER IN THIS BOX

(d) List four (04) weaknesses of RPC.

(4 marks)

ANSWER IN THIS BOX

(e) Define a Message Broker in the context of middleware architectures.

(5 marks)

ANSWER IN THIS BOX

4.

(a) What are the consequences of not having IDL in *message queueing* (message oriented middleware)?

(5 marks)

ANSWER IN THIS BOX

(b) Identify the most suitable **HTTP verb** to be used in each of the following RESTful URIs.

(5 marks)

ANSWER IN THIS BOX

URI	HTTP Method
/coffeOrder/{id}	
/coffeOrder/add	
/coffeOrder/delete/{id}	
/coffeOrder/getAll	
/coffeOrder/update/{id}	

(c) The following piece of code was taken from a **Controller class** in a Restful backend application.

Explain The *functionality* of the code given below.

(10 marks)

```
package com.example.coffeeapp.controller;

import com.example.coffeeapp.model.CoffeeOrder;
import com.example.coffeeapp.service.CoffeeOrderService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/orders")
public class CoffeeOrderController {
```

```
    @Autowired
    private CoffeeOrderService coffeeOrderService;

    @PostMapping
    public ResponseEntity<CoffeeOrder> createOrder(@RequestBody CoffeeOrder
    order) {
        CoffeeOrder createdOrder = coffeeOrderService.createOrder(order);
        return ResponseEntity.ok(createdOrder);
    }

    @GetMapping
    public ResponseEntity<List<CoffeeOrder>> getAllOrders() {
        return ResponseEntity.ok(coffeeOrderService.getAllOrders());
    }


    @GetMapping("/{id}")
    public ResponseEntity<CoffeeOrder> getOrderById(@PathVariable Long id) {
        CoffeeOrder order = coffeeOrderService.getOrderById(id);
        return order != null ? ResponseEntity.ok(order) :
    ResponseEntity.notFound().build();
    }


    @PutMapping("/{id}")
    public ResponseEntity<CoffeeOrder> updateOrder(@PathVariable Long id,
    @RequestBody CoffeeOrder updatedOrder) {
        CoffeeOrder order = coffeeOrderService.updateOrder(id, updatedOrder);
        return order != null ? ResponseEntity.ok(order) :
    ResponseEntity.notFound().build();
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteOrder(@PathVariable Long id) {
        boolean deleted = coffeeOrderService.deleteOrder(id);
        return deleted ? ResponseEntity.noContent().build() :
    ResponseEntity.notFound().build();
    }
}
```

ANSWER IN THIS BOX

(d) The following service call was found in a piece of code in a **Controller class** of a Restful backend application.

```
Order order = orderService.findOrderById(orderId);
```

Briefly explain the expected functionality of the above *service call*.

(5 marks)

ANSWER IN THIS BOX

*****